



roundabout

Ver.2

ユーザマニュアル

開発リファレンス

第 6 版

最終更新日 2012/7/26

目次

第 1 章	はじめに.....	8
1-1	HTML の記述.....	8
1-2	HTML5 での記述.....	9
1-3	スタイルシートの利用.....	9
(1)	i モードブラウザ 1.0 搭載端末への対応.....	9
(2)	利用する際の制限事項.....	10
1-4	絵文字の利用.....	10
(1)	対象となる絵文字.....	10
(2)	利用する際の制限事項.....	10
第 2 章	画像変換.....	12
2-1	変換画像の基準幅.....	12
2-2	画像変換対象となるファイル.....	12
2-3	画像サイズ自動変換.....	13
(1)	画像幅に合わせた縮小変換.....	13
(2)	ブラウザキャッシュ容量に合わせた自動容量調整.....	13
(3)	画像を変換してもブラウザキャッシュ容量内に収まらない場合の代替処理.....	13
2-4	アニメーション GIF.....	13
2-5	オートレイアウト.....	13
2-6	画像個別の画像変換指示.....	14
(1)	ra:width 属性の追加.....	14
(2)	ra:size 属性の追加.....	15
(3)	ra:convert 属性の追加.....	16

(4)	背景画像の変換指示	16
(5)	画像サイズ絶対値指定の上書き	16
2-7	非対応端末向けのフォーマット変換	17
2-8	用意した画像の出し分け	18
2-9	画像の転送禁止設定	18
2-10	絶対 URL で指定した画像の変換	18
(1)	HTML と画像を同一サーバから配信している場合	19
(2)	HTML と画像を異なるサーバから配信している場合	19
(3)	CDN 経由で画像を配信している場合	20
2-11	パラメータによる動的画像の変換(ラウンドアバウト ビヨンドのみ)	20
2-12	viewport 機能を利用したオートレイアウト	21
(1)	サーバ側で縮小変換しない場合	21
(2)	ra:width 属性が指定された画像のみ縮小変換する場合	22
第 3 章	端末情報ファイル	23
3-1	端末情報ファイルの用途	23
3-2	利用可能な端末情報ファイルの項目	23
3-3	端末情報のリクエストヘッダ出力	23
3-4	非対応端末設定ファイル	23
3-5	非対応端末の制御	24
第 4 章	端末グループ	25
4-1	端末グループの種類	25
4-2	端末グループのネーミングルール	25
4-3	ユーザー端末グループの定義	25
4-4	初期設定済のユーザー端末グループ	26

4-5	ユーザー端末グループの利用	27
4-6	クローラーグループの定義	27
4-7	クローラーグループの利用	28
4-8	グループ設定の再読み込み	28
第 5 章	表示コントロール(PI)	29
5-1	利用可能な端末情報	29
5-2	PI で使用できる書式	30
(1)	一致	30
(2)	範囲指定	30
(3)	否定	30
(4)	OR	31
(5)	AND	31
(6)	ユーザー拡張項目へのアクセス方法	32
(7)	複数行にわたる PI	32
(8)	タグ表記	33
第 6 章	テーブル変換	35
6-1	タイプ1(リスト表現)	35
6-2	タイプ2(メニュー表現)	36
6-3	タイプ3(回り込み表現)	37
6-4	補足	39
(1)	a タグ内のブロック要素変換	39
第 7 章	CSS 変換シート	40
7-1	CSS 変換シートの配置場所	40
7-2	CSS 変換シートの内容	40
7-3	CSS 変換ルール適用の優先度	41

第 8 章	CSS・JavaScript ファイル切り替え.....	42
8-1	CSS ファイル切り替えの例	42
8-2	JavaScript ファイル切り替えの例.....	43
8-3	ra:rename 属性の書式	43
第 9 章	HTML 変換シート	46
9-1	HTML 変換シートの種類.....	46
(1)	デフォルト言語変換シート	46
(2)	絵文字変換シート	46
(3)	独自言語変換シート	46
9-2	HTML 変換シートの書式.....	46
9-3	HTML 変換シートにおける端末グループ	47
9-4	独自言語変換シートの作成・編集	47
9-5	HTML 変換シートの適用.....	48
9-6	HTML 変換シートの再読み込み	49
第 10 章	キャラクタエンコーディング変換.....	50
10-1	変換対象のキャラクタエンコーディング	50
10-2	使用可能な文字セット.....	50
10-3	Content-Type 変換.....	50
10-4	キャラクタエンコーディングの指定	51
第 11 章	付録	52
11-1	端末情報ファイルの内容 (terminfo2.csv)	52
11-2	初期設定済のユーザー端末グループ (device-group.conf).....	56

改訂履歴

版数	発行日	改訂内容
第 1 版	2011 年 5 月 24 日	初版発行
第 2 版	2011 年 6 月 8 日	<ul style="list-style-type: none"> ・目次の項番「()」以下を省略 ・一部見出しを変更(内容への変更無し) ・ソース、設定ファイルの表記を変更(行番号追加、等幅フォント、一部記号の半角表記、フォントカラーではなく書体を用いた強調、フキダシを削除) ・1 章 1-1 (1) を削除し、1 章 1-1 と統合 ・1 章 1-2 (1) を削除し、1 章 1-2 と統合 ・2 章 2-3 (4) を削除し、2 章 2-3 (3)と統合 ・10 章 10-1 の表の空欄に斜線追加、同一内容のセルの結合
第 3 版	2011 年 6 月 24 日	<ul style="list-style-type: none"> ・10 章 10-2 ブラウザ小サイズ端末のデバイスグループ定義を追加
第 4 版	2011 年 9 月 15 日	<ul style="list-style-type: none"> ・1 章 1-2 「HTML5 での記述」追加 ・2 章 2-6(2) 「絶対 URL で指定した画像の変換」を設定していた時の仕様を追加 ・2 章 2-9 スマートフォンの場合の制限を追加 ・2 章 2-10 を「絶対 URL で指定した画像の変換」に変更し、3 つの設定例を追加 ・2 章 2-12 viewport 機能を利用したオーレイアウトを追加 ・7 章 JavaScript ファイル切り替えについての説明を追加 ・10 章 10-2 携帯全般のデバイスグループと WindowsMobile のデバイスグループ定義を追加

第 5 版	2012 年 7 月 13 日	<ul style="list-style-type: none">・1 章 1-3 複数クラス指定に関する記述、および制限事項を追加・2 章 2-6(5)「画像絶対値指定の上書き」を追加・2 章 2-7 24bitPNG に関する記述を追加・6 章 「テーブル変換」を追加
第 6 版	2012 年 7 月 26 日	<ul style="list-style-type: none">・1 章 1-2 HTML5 タグの変換ルールを追加

第1章 はじめに

本書は、ラウンドアバウトの機能全般のリファレンスになります。ラウンドアバウトを初めてお使いになる場合には、まずは、別紙「ラウンドアバウト 2 スタートガイド」をお読みいただくと開発がスムーズに始められます。ラウンドアバウトをカスタマイズして使うなど、機能の詳細について知りたい場合は本書をお読みください。

- ※ 文中の各種ファイルについて、配置ディレクトリなどはパッケージに同梱の「README.html」に詳細が記載されていますので、そちらを参照ください。
- ※ 文中には、Apache の設定に係わる記載がありますが、こちらについての詳細は、別紙「Apache 設定リファレンス」で解説されていますので、そちらを参照ください。

1-1 HTML の記述

通常の XHTML/HTML で必要なものを入れられます。ラウンドアバウトを使う上でポイントとなる以下の 4 つの宣言、meta について解説します。

01	<?xml version="1.0" encoding="UTF-8"?>
02	<?ra-page viewport="auto" image-convert="no" g="SP" ?>
03	<meta name="disparea" content="vga" />
04	<meta name="viewport" content="width=device-width, initial-scale=0.666, maximum-scale=0.666, minimum-scale=0.666" />

- ① ラウンドアバウトでは XHTML を標準として推奨していますので始めに XML 宣言を書きます。(1 行目)
- ② 画像変換を行わない場合に、スマートフォンの viewport 設定に最適値を与えるラウンドアバウトの機能です。ラウンドアバウトでは VGA ベースのコンテンツ作成がデフォルト設定で、スマートフォンには標準解像度である HVGA サイズに変換して出力します。画面の大きいスマートフォンでは、画像変換を行うと画像が劣化して見えるため、作成した VGA ベースの画像をそのまま出したい場合には、この指定を入れてください。XML 宣言より後、HTML タグ出現前に書きます。この設定を入れるとランドスケープビュー時でも最適な表示になります。通常の viewport 設定を併記した場合、ラウンドアバウトが有効であれば、この「ra-page」が有効となります。(2 行目)
- ③ i モードブラウザ 2.x で、VGA モードを使用する指定です。ラウンドアバウトは i モードブラウザ 2.x 端末を VGA 端末として扱いますので、この設定は必ず入れてください。(3 行目)
- ④ 通常の viewport 設定です。上記は、VGA ベースでコンテンツを作成し、ラウンドアバウトを通さな

い場合に適切な設定例です。②を併記してラウンドアバウトが有効な場合は、こちらは無効となり②が有効となります。(4行目)

※ iモードブラウザ 1.0 搭載端末に対しては、1行目が XML 宣言となっている場合、コンテンツが XHTML と解釈され、Content-Type が application/xhtml+xml に自動設定されます。

1-2 HTML5 での記述

ラウンドアバウト 2.1.0 から、HTML5 の一部 (DOCTYPE 宣言、meta 要素の charset 属性、input 要素に追加された新しい type 属性) の言語変換に対応し、HTML5 でのページ制作も可能です。ただし、変換をする一部の記述以外は、HTML5 に対応していない携帯端末含めどの機種にもそのまま表示がされます。バージョン 2.2 以降では、HTML5 に対応していない携帯端末では、div や span など無害なタグに変換されます。

なお、HTML5 でページを作成する場合、XHTML ならば必要な XML 宣言 (1-1「HTML の記述」①) を記述する必要はありません。

1-3 スタイルシートの利用

ラウンドアバウトでは外部 CSS の内容を一部の機種向けに style 属性へとインライン展開します。また、CSS 変換機能 (第 6 章「CSS 変換シート」参照) は外部 CSS ファイルと連動して機能します。これらの機能を使うために、CSS は外部ファイルとして用意してください。style タグ、style 属性を使った場合は、前述の機能は使えませんので注意してください。

CSS の記述方法は、通常のスタイルシートの文法に従って記述しますが、一部制限事項があります。@規則はサポートしていませんので、複数ファイルを読み込む場合は link タグを使用します。

(1) iモードブラウザ 1.0 搭載端末への対応

外部 CSS に対応していない iモードブラウザ 1.0 搭載端末 (900 シリーズ、および、それ以前に発売の機種) には、外部 CSS の内容が style 属性にインライン展開されます。

iモードブラウザ 1.0 で使用できるプロパティは、font-size、color、background-color です。このプロパティを効かせるには、div か span タグに class 属性や id 属性を指定して、外部 CSS に設定値を書きます。CSS の書き方は ID セレクタもしくはクラスセレクタのどちらかで指定します。

例: html ファイル

01	<div class="foo"> ... </div>
----	------------------------------

例: css ファイル

```
01 .foo { font-size:small; color: #EEEEEE; background-color: #333333; }
```

(2) 利用する際の制限事項

ラウンドアバウト 2.1 以下では以下の制限事項があります。指定を行った場合、CSS 展開機能は動作しません。

- HTML タグに id 属性と class 属性を両方指定することはできません
- class 属性に複数のクラスを指定することはできません
- style 属性と id/class 属性を同時に指定することはできません

また CSS ファイルには、以下の制限事項があります。

- @規則 (@import など) が使用できません
- 指定可能なセレクトタは、ID セレクトタ、クラスセレクトタ、タイプセレクトタ (body, td, th のみ) と疑似セレクトタ (a:link, a:visited, a:active のみ) のいずれかのみです
- ".foo.bar { ... }" のように複数のセレクトタを複合的に指定することはできません (指定した場合は、CSS 展開機能が機能しません)

1-4 絵文字の利用

(1) 対象となる絵文字

絵文字は下記のいずれかを指定します。

- docomo 絵文字
 - 16 進数表記
 - 10 進数表記
 - Shift_JIS コード出力

※ SoftBank 絵文字 16 進数表記、au 絵文字 アイコン番号表記からの変換はオプションです。これらの絵文字表記から変換したい場合は、開発者サイト「ラウンドアバウト デベロッパーコネクション (<http://developer.symmetric.jp/roundabout/>)」より変換シートをダウンロードしてお使いいただけます。

(2) 利用する際の制限事項

HTMLに含まれるすべての絵文字が変換対象となるため、title要素や属性内に記述する絵文字は正しく変換・表示できない場合があります。

16進数表記時の英字は小文字で記述してください。

第2章 画像変換

2-1 変換画像の基準幅

ラウンドアバウトの画像変換は、ページ内のどんな大きさの画像も、解像度の違う端末で見ても同じ比率で見えるように変換します。その変換のためには、個々の画像が、何ピクセルの解像度の画面を想定して作られているのかを設定しておく必要があります。

デフォルト設定は、タブレット端末を除く、現在のモバイル端末の最高スペックとなる、VGA 画面の幅、480 ピクセルになっています。

2-2 画像変換対象となるファイル

ラウンドアバウトの画像変換対象となるファイルは、以下のタグや属性により HTML 上から参照された画像で、かつウェブサーバのファイルシステム上からアクセス可能な画像ファイルです。

画像の種類	記述方法
インライン画像	・img タグの src 属性
背景画像	・body タグの background 属性 ・body, div, p, table, tr, th, td, h1~h6, ol, ul, li, span タグの style 属性で指定した background および background-image プロパティ ・外部 CSS 内で指定した background・background-image プロパティ
イメージボタン	・input タグの src 属性
リストのマーカ画像	・li タグの style 属性で指定した list-style-image プロパティ ・外部 CSS 内で指定した list-style-image プロパティ
画像リンク	・a タグの href 属性

※ 1つのタグや属性に上記画像を2つ以上組み合わせて指定することはできません。

一般的なウェブサーバの構成で HTML と画像が同一サーバ上に存在する場合、ラウンドアバウトが有効なディレクトリ以下の、すべての画像が変換対象となります。

また、絶対 URL で画像を指定した場合は、ラウンドアバウトがインストールされたウェブサーバが画像を配信している場合にも、変換の対象となります。詳しくは[画像が異なるサーバに存在する場合](#)を参照してください。

2-3 画像サイズ自動変換

(1) 画像幅に合わせた縮小変換

元画像の横幅がアクセスした端末の画面幅(ブラウザの可視領域)に収まりきらない場合は、可視領域に収まるように自動的に画像を縮小変換し、出力します。

(2) ブラウザキャッシュ容量に合わせた自動容量調整

全画像を含めた HTML サイズが端末のブラウザキャッシュ容量に納まるように、JPEG 高圧縮化・パレット数削減による画像容量の調整を行います。

(3) 画像を変換してもブラウザキャッシュ容量内に収まらない場合の代替処理

画像を最低品質まで劣化させても端末のブラウザキャッシュ容量内に収まらない場合、ラウンドアバウトはスペーサ画像もしくは代替文字列に置き換えて出力します。スペーサ画像は1ピクセルx1ピクセルのGIF画像(GIFに対応していない端末はPNG)です。ページ全体のレイアウトを保持するため、変換後の画像と同じ大きさとなるようimgタグにwidth・height属性を追加し、スペーサ画像を引き伸ばします。代替文字列は、alt属性が使用されます。

なお、スペーサ画像に変換せずに代替文字列を優先して表示したい場合は roundabout-vhost.conf で設定が可能です。詳しくは別紙「Apache 設定リファレンス」の RALCSpacerImage を参照ください。

2-4 アニメーション GIF

アニメーション GIF はローカルカラーマップのアニメーション GIF に変換されます。また、元画像のアニメーション GIF のフレーム数が5フレームを超える画像の場合は、最初の1フレーム目だけが出力されます。

GIF形式の画像に対応していない端末には、最初の1フレーム目の画像がPNGに変換され出力されます。

※ アニメーション GIF は各フレームの縦横サイズが全て同一である必要があります。画像編集ソフトによっては、デフォルト設定が各フレームの差分データしか持たないようになっていますので注意が必要です。

2-5 オートレイアウト

ラウンドアバウトを利用した開発では基準となる画面幅を決めた上での開発を推奨しています。ラウンドアバウトでは、決められた基準から決められたルールで自動的に画像を変換し、すべての端末で表

示できるように変換します。例えば、基準となる画面幅を VGA と決めた場合、QVGA 画面幅の端末に対しては、縮尺比 0.5 に画像を自動変換し、出力します。

ラウンドアバウト設定例:

01	RALCBaseWidth 480
02	RALCImageScaleList 96, 120, 225, 228, 230, 232, 234, 240, 314, 320, 468, 480

上記の例は基準画面幅を VGA にした場合の設定となります。オートレイアウトの設定の詳細に関しては、別紙「Apache 設定リファレンス」RALCBaseWidth、RALCImageScaleList 項を参照ください。

オートレイアウト機能はデフォルトでは有効で、サイト全体に適用されています。設定により無効にすることも可能です。画像を個別にオートレイアウトさせることも可能ですので、全体の設定を無効にすることで、よりユーザーの意思に近く、自由度の高い表現が可能となります。

2-6 画像個別の画像変換指示

オートレイアウトでは、ページ全体・サイト全体の画像に適用されますが、画像個別に変換指示を設定することも可能です。そのためには、img タグや body タグにラウンドアバウト固有の属性を追加します。

(1) ra:width 属性の追加

① オートスケールの有効/無効を指定する

画像に対し、個別にオートスケールを有効にする場合は以下のように指定します。

例:

01	<code></code>
----	--

また、逆に特定の画像に対し、オートスケールを無効にする場合には以下のように指定します。

例:

01	<code></code>
----	---

※ ra:width 属性はインライン画像 (img タグ) および画像リンク (a タグ) にのみ有効です。

② 画像幅の相対値指定

サムネイル画像など、大きな画像を小さく使いたい場合で、かつ、どの解像度の端末で見ても画面サイズに対して同じ比率の大きさで見えるようにしたい場合は、`ra:width` 属性で「r」という単位をつけて指定します。

例えば、200 ピクセルある画像を基準幅の端末で 80 ピクセルに縮小した状態で表示し、他の解像度の端末でも同じ大きさに見えるようにするには、以下のように指定します。

例:

```
01 
```

- ※ 画像の大きさが指定した値より小さい場合、拡大はしません。
- ※ バイト数が大きすぎる画像は変換対象外となりますので注意してください。変換対象とするバイト数を変更することもできます。詳しくは「Apache 設定リファレンス」を参照ください。

③ 画像幅の絶対値指定

画像を、どの解像度の端末で見ても同じピクセル数で出力をさせたい場合は `ra:width` 属性で指定を行います。指定したい横幅ピクセル数を、単位をつけずに書きます。画像を横幅 120 ピクセルで出力させたい場合は以下のように指定します。

例:

```
01 
```

(2) `ra:size` 属性の追加

ラウンドアバウトでは容量計算の際に、ローカルに存在するファイルを直に参照してファイル容量を取得します。しかし、外部ファイル(絶対 URL で指定されたファイル)の容量は把握することができません。その場合は、`ra:size` 属性を利用し、ファイルサイズを通知することで、より精度の高い容量計算を行うことが可能になります。

- ※ ただし絶対 URL で指定した画像の変換で変換設定が正しく指定された場合には、ファイル容量が自動で取得されますので、`ra:size` 属性を指定する必要がありません。

例:

```
01 
```

上記では、画像を 1024 バイトと指定します。この指定で、ラウンドアバウトは端末が持つコンテンツ表示許容量から 1024 バイトを差し引いた容量で容量計算を行います。

画像以外では、FlashやCSSもra:size属性を利用することで容量の通知が可能です。

ra:size 属性は、画像を指定できる全てのタグと link、object、script タグで使用できます。

例:

```
01 <object ra:size="12345" data="http://example.jp/example.swf" ...
```

(3) ra:convert 属性の追加

コンテンツの中には、肖像権がある画像や著作権の事情で画像を変換できない場合があります。その場合、以下のように指定することで画像変換対象から外すことが可能です。

例:

```
01 
```

指定された画像は、完全に画像変換対象から外されます。但し、自動容量調整の容量計算のために、画像の容量は計算対象に含まれます。

(4) 背景画像の変換指示

CSS で指定する背景画像についても、ra:width 属性、ra:size 属性、ra:convert 属性が使えます。背景画像に変換指示を行うには、背景画像を含んだセクタ内に、ラウンドアバウト拡張のプロパティ名を使います。値は、属性の場合と同じものが使えます。

属性名	拡張プロパティ名
ra:width	-roundabout-width
ra:size	-roundabout-size
ra:convert	-roundabout-convert

例:

```
01 .foo {
02 background-image: url(a.jpg);
03 -roundabout-width: 120r;
04 }
```

(5) 画像サイズ絶対値指定の上書き

ラウンドアバウト 2.2.0 から、img タグの width 属性・height 属性で絶対値指定した画像サイズを上書きすることができます。

従来ラウンドアバウトでは、元となる HTML で画像サイズが絶対値指定されていた場合、画像変換を行ってもブラウザ側でリサイズされて表示されてしまうため効果がありませんでした。ラウンドアバウト 2.2.0 以降では絶対値指定された値をラウンドアバウトで変換した画像の幅・高さで上書きします。

例:元の HTML

```
01 
```

例:変換後の HTML

```
01 
```

当機能は RALCImgAttributeOverwrite ディレクティブで On/Off を設定します。

ラウンドアバウト設定例:

```
01 RALCImgAttributeOverwrite On
```

- ※ 上書き対象となる設定は img タグの width 属性・height 属性で絶対値指定したもののみです。相対値(パーセント)指定のものは上書きしません。
- ※ 当機能を使用する場合、スペーサー画像など width 属性・height 属性を上書きされては困る画像には ra:convert="no"を使用する必要があります。

2-7 非対応端末向けのフォーマット変換

用意した画像フォーマットに対応していない端末からのアクセス時には、自動的に端末で表示可能な画像フォーマットに変換して出力します。

変換対象フォーマット: JPEG、GIF、8ビット PNG、24ビット PNG(アルファチャネルなし)

- ※ 8ビット PNG は GIF に、24ビット PNG は JPEG に変換されます。ただし、ラウンドアバウト 2.2 以降では PNG へのリンク画像はすべて JPEG に変換されます。
- ※ BMP、インターレース画像、プログレッシブ JPEG、アルファチャネル付きの PNG はサポートしていません。
- ※ ラウンドアバウト 2.1 以下では、24ビット PNG に対応していません。

2-8 用意した画像の出し分け

肖像権や著作権の事情などで画像を自動変換させたくない場合は、予め段階的に用意された画像（例：大中小）をアクセスした端末のスペック（横幅ピクセル数）で出し分ける事ができます。

設定例 (roundabout-vhost.conf) :

※ 以下は roundabout-vhost.conf の初期設定と同じです。

01	RALCImgFileDispatch	1-139	_XS
02	RALCImgFileDispatch	140-219	_S
03	RALCImgFileDispatch	220-399	_M
04	RALCImgFileDispatch	400-799	_L
05	RALCImgFileDispatch	800-3000	_XL

HTML 記述例:

01	<code></code>
----	---

上記の例では、拡張子を除いたファイル名の末尾が RALCImgFileDispatch で指定した末尾“_L”と一致するため、画像切り替えの対象となります。ラウンドアバウトが動作している環境下では、QVGA 端末のアクセス時には末尾“_M”が使用されるため、画像 example_M.jpg が表示されます。

なお、RALCImgFileDispatch を利用する場合はオートレイアウト機能を OFF、また画像に無変換指定 (ra:convert="no") を行ってください。オートレイアウト機能が ON の場合、画像切り替え後にオートレイアウトがかかってしまい、予期しない変換となる場合があります。

設定の詳細は、別紙「Apache 設定リファレンス」の RALCImgFileDispatch 項を参照ください。

2-9 画像の転送禁止設定

docomo、au、SoftBank の 3 キャリアに転送禁止フラグを設定することが可能です。画像を転送禁止にするには、画像のコメント領域に以下の文字列を埋め込みます。

01	<code>kddi_copyright=on, copy="NO"</code>
----	---

※ SoftBank には、画像の Content-Type を Forward Lock 方式の DRM Message の型にしているため、PC ブラウザでは画像が表示されない場合があります。

2-10 絶対 URL で指定した画像の変換

画像 URL を絶対 URL で指定した場合は、HTML を配信するサーバと画像を配信するサーバの両方に

ラウンドアバウトがインストールされている場合のみ、画像変換を行うことができます。絶対 URL で指定された画像の変換を行うには、次のような設定を行います。

(1) HTML と画像を同一サーバから配信している場合

以下の例では、HTML と画像をドメイン `www.example.jp` で配信している場合を想定しています。
RALCUrlMapping ディレクティブで対象のドメインとローカルへのマッピングパス(/)を指定します。

設定例 (roundabout-vhost.conf) :

01	RALCUrlMapping www.example.jp / /
----	-----------------------------------

HTML 記述例:

01	
----	---

HTML と画像が別ドメインの場合も、RALCUrlMapping で対応できます。

※ RALCUrlMapping ディレクティブの設定は、CSS や外部 JavaScript、Flash などのコンテンツ取得時にも利用されます。

(2) HTML と画像を異なるサーバから配信している場合

HTML と画像ファイルが異なるサーバ上に存在している場合、画像をローカルにマウントするなどの方法により、`mod_ralc` 側から画像ファイルが参照できる環境であれば画像変換を行うことができます。

以下の例では、HTML サーバ `www.example.jp` と画像サーバ `img.example.jp` とで構成される環境で、画像サーバ上のドキュメントルートが HTML サーバ上のパス `/mnt/img/htdocs` にマウントされている場合を想定しています。

設定例 (HTML サーバ側 roundabout-vhost.conf) :

01	RALCUrlMapping img.example.jp / /localimage/
02	Alias /localimage/ /mnt/img/htdocs/

HTML 記述例:

01	
----	--

※ RALCUrlMapping ディレクティブの設定は、CSS や外部 JavaScript、Flash などのコンテンツ取得時にも利用されます。

(3) CDN 経由で画像を配信している場合

画像を CDN 経由で配信している場合、CDN のオリジンサーバが HTML 配信サーバと同一サーバかどうかにより、前述(1)もしくは(2)の設定を行います。

たとえば、CDN の URL が `cdn.example.net` で、HTML と画像ファイルが同ドメイン `www.example.jp` から配信を行っている環境での設定を示します。元画像は `http://www.example.jp/images/logo.jpg` よりアクセスできるものとします。

設定例 (roundabout-vhost.conf) :

01	<code>RALCUrlMapping cdn.example.net /foo/ /</code>
02	<code>RALCImageSizeRounding type1</code>

HTML 記述例:

01	<code></code>
----	---

CDN を利用する場合は、`RALCImageSizeRounding` ディレクティブを指定し、ラウンドアバウトが画像に自動付与するパラメータのパターンを限定化することをお勧めします。設定の詳細は、別紙「Apache 設定リファレンス」の `RALCImageSizeRounding` 項を参照ください。

※ `RALCUrlMapping` ディレクティブの設定は、CSS や外部 JavaScript、Flash などのコンテンツ取得時にも利用されます。

2-11 パラメータによる動的画像の変換(ラウンドアバウト ビヨンドのみ)

パラメータも含めて 1 つの画像として扱うことで、パラメータの値によって動的に切り替わる画像(地図画像など)でも画像変換を行うことができます。

含めたいパラメータ名を `ImageApplicationParam` ディレクティブにて設定します。複数設定できます。以下の例では、`id`、`site`、`cat` のパラメータ名を動的画像のキーに設定しています。

例:

01	<code></code>
----	---

下記、①と②は、同じ「image.gif」でも、パラメータが異なるため、別の画像として扱われます。

image.gif?id=0123456&site=5&cat=1 …①

image.gif?id=9876543&site=7&cat=3 …②

また、動的画像のキーとなるパラメータの指定を、同じ画像のパラメータ群の中にも含めることもできます。キーとなるパラメータを、とあるパラメータ名の値にすることで、この値を書き換えることで、キーとなるパラメータの増減に即座に対応することができます。設定は ImageApplicatioinParamByQuery ディレクティブにて行います。

以下の例では、動的画像のキーとなるパラメータをパラメータ値として持たせるためのパラメータ名を「_iap」と設定し、id、site、cat のパラメータ名を動的画像のキーに設定しています。

例:

01	
----	--

さらに、パラメータ名 stat を動的画像のキーに追加する場合は、以下のようになります。

01	
----	---

詳しい設定方法については、別紙「Apache 設定リファレンス」の ImageApplicatioinParam、ImageApplicatioinParamByQuery 項を参照ください。

2-12 viewport 機能を利用したオートレイアウト

スマートフォンではブラウザの viewport 機能を利用したオートレイアウト(縮小変換)を行うことができます。viewport 変換機能によるオートレイアウトのメリットはサーバ側で画像変換を行わずに画像の大きさを調整できるため、クオリティの高い元画像をそのままスマートフォンに配信できる点にあります。

(1) サーバ側で縮小変換しない場合

viewport 機能によるオートレイアウトを使用し、サーバ側で画像の縮小変換を行わないようにするには、次のようにページレベルの PI(2 行目)を指定します。

例:

01	<?xml version="1.0" encoding="UTF-8"?>
02	<?ra-page viewport="auto" image-convert="no" g="SP"?>
03	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

viewport 属性には auto を指定し、image-convert 属性には no を指定します。g 属性には適用する端末グループを指定します。viewport 機能は iPhone/Android/Windows Phone に対応しているため、スマートフォン以外の端末グループを指定することはできません。

上記の PI を記述すると、viewport を指定した meta タグが自動的に挿入されます。

(2) ra:width 属性が指定された画像のみ縮小変換する場合

ra:width 属性が指定された画像のみ、サーバ側で画像の縮小変換を行い、それ以外の画像は元画像をそのまま利用する場合には、次のようにページレベルの PI(2 行目)を指定します。

例:

01	<?xml version="1.0" encoding="UTF-8"?>
02	<?ra-page viewport="auto" image-convert="ra:width" g="SP"?>
03	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

viewport 属性には auto を指定し、image-convert 属性には ra:width を指定します。g 属性には適用する端末グループを指定します。viewport 機能は iPhone/Android/Windows Phone に対応しているため、スマートフォン以外の端末グループを指定することはできません。

上記の PI を記述すると、viewport を指定した meta タグが自動的に挿入されます。

第3章 端末情報ファイル

ラウンドアバウトではすべての端末の詳細なスペック情報を内部のデータベースとして持っています。画像変換(第2章参照)や表示コントロール(PI)(第5章参照)はすべてこの情報を元に動作しています。

3-1 端末情報ファイルの用途

ラウンドアバウトの画像変換や言語変換では、より端末に適したコンテンツを出力するため、端末情報ファイルを利用します。また、端末情報はラウンドアバウトが利用するのみならず、バックエンドに配備されるアプリケーションでも利用することが可能です。

3-2 利用可能な端末情報ファイルの項目

第10章「付録」の「[端末情報ファイルの内容](#)」を参照ください。

3-3 端末情報のリクエストヘッダ出力

端末情報はリクエストヘッダとして PHP などのアプリケーションに渡されます。アプリケーション側では、リクエストヘッダから端末情報を取得することにより、機種情報を利用した処理を行うことができます。

端末情報ヘッダは、X-RA-(項目名): (値) の形式で追加されます。端末グループのように値が複数となる場合は、複数個のリクエストヘッダが追加されます。端末情報ヘッダは HTTP ヘッダの一般的な書式に則り、各単語の先頭は大文字で送出されます。

端末情報ヘッダの例:

01	X-RA-Device-Name: N905
02	X-RA-Screen-Height: 400
03	X-RA-Device-Group: DCM
04	X-RA-Device-Group: L2M

追加される端末情報の項目は、RALCTerminfoHeader ディレクティブで指定します。

3-4 非対応端末設定ファイル

ラウンドアバウトではキャッシュサイズが低すぎるなど、端末依存の問題で正しく表示を行うことができない可能性の高い端末を非推奨端末としています。これらの非推奨端末は、非対応端末設定ファイル (unsupport.csv) のデフォルト値となっています。

非対応端末設定ファイルはユーザーが自由に変更可能です。非対応対象機種を追加する場合は、端末情報ファイル (terminfo.csv) に記載の Device-Id または Carrier または端末グループ名を非対応端末設定ファイルに追加してください。端末グループ名を指定する場合は、先頭に@ (アットマーク) をつけてください。

※ 端末グループについては、第 4 章「[端末グループ](#)」を参照ください。

3-5 非対応端末の制御

ラウンドアバウトでは非対応端末の制御方法として、以下の方法を提供しています。

- (A) 非対応端末向けページへのリダイレクト (非対応端末からアクセスがあった場合)
- (B) 非対応端末かどうかを示すリクエストヘッダ X-RA-Unsupport の設定

非対応ページへのリダイレクトを使用する場合、RALCUnsupportUrl ディレクティブでリダイレクト先 URL を指定します。

リダイレクト機能を使用しない場合は、X-RA-Unsupport ヘッダをアプリケーションから参照することにより、非対応端末かどうかを判定して処理を行うことができます。

非対応端末からアクセスしたときのリクエストヘッダ:

```
X-RA-Unsupport: 1
```

対応端末からアクセスしたときのリクエストヘッダ:

```
X-RA-Unsupport: 0
```

第4章 端末グループ

端末グループは、いくつかの端末をグルーピングしたものです。端末グループは XHTML や CSS 出し分けの判定条件として利用したり、端末情報ヘッダとしてリクエストヘッダに追加したりできます。

4-1 端末グループの種類

端末グループにはユーザー端末グループ・クローラーグループがあり、ユーザー定義が可能です。テキスト形式の設定ファイルです。

4-2 端末グループのネーミングルール

グループの名称は、ユーザーが自由に定義できます。使用可能な文字は半角英数字のみで、記号は使用できません。また、数字のみのグループ名は使用できません。

ユーザー端末グループ名の例：

- IPHONE
- DCM
- KDDI

クローラーグループ名の例：

- crawlerGoogle
- crawlerYahoo

使用できない名前例：(数字のみ、記号が入っている)

- 1234567890
- crawler?

4-3 ユーザー端末グループの定義

ユーザー端末グループは、端末グループ設定ファイル(device-group.conf)で定義します。

端末グループ設定ファイルはテキスト形式のファイルです。端末グループ設定ファイルには、ユーザー端末グループ名とその条件をセットで記述します。条件には 1 つ以上の PI 表記の条件式を記述します。

※ 条件式の書き方については第 5 章「[表示コントロール\(PI\)](#)」を参照ください。

書式:

01	#1.0.0
02	[ユーザー端末グループ名]{
03	[PI 表記の条件式]
	:
04	}

※ 1 行目は「1.0.0」固定です。

※ 「#」から始まる行はコメント行です。

{ }内にはPI表記の条件式を1つ以上記述します。2つ以上の条件式を指定した場合、いずれかの条件式にマッチすればそのユーザー端末グループに所属すると判断されます。

同一名のユーザー端末グループを記述した場合、ファイル内で最初に出現したものが優先されます。

定義したグループ名を、別のグループ定義に利用することができます。その場合、参照されるグループ名は、利用する箇所より前で定義されている必要があります。

例:

01	#1.0.0
02	#スマートフォン
03	SP {
04	<?ra ot="200-299" bt="200-899"?>
05	}
06	#タブレット型
07	TABLET {
08	<?ra g="SP" Screen-Inch="7.0-50.0"?>
09	}

※ 「ot」は「Os-Type」、「bt」は「Browser-Type」、「g」は「Device-Group」の略記です。詳細は第 10 章「付録」の「[端末情報ファイルの内容](#)」を参照ください。

4-4 初期設定済のユーザー端末グループ

端末グループ設定ファイルには、予め最も基本となる端末グループが定義されています。こちらについ

ては第 10 章「付録」の「[初期設定済のユーザー端末グループ](#)」を参照ください。

4-5 ユーザー端末グループの利用

ユーザー端末グループを利用するには、RALCDeviceGroupFile ディレクティブで端末グループ設定ファイルを指定します。

RALCDeviceGroupFile ディレクティブは、<Location>ディレクティブ内でも指定可能です。

例:

01	<Location /foo/>
02	RALCDeviceGroupFile /var/roundabout/conf/device-group.conf
03	</Location>

PIでの利用方法については、第 5 章「[表示コントロール\(PI\)](#)」を参照ください。

4-6 クローラーグループの定義

クローラーグループは、クローラーグループ設定ファイル(crawler-group.csv)で定義します。

クローラーグループ設定ファイルは CSV 形式のテキストファイルです。クローラーグループ設定ファイルには、クローラーグループ名とマッチングさせる User-Agent のパターンをペアで指定します。ラウンドアバウトは HTTP リクエストの User-Agent ヘッダと User-Agent パターンとのマッチングを行い、マッチしたクローラーグループを端末グループに設定します。

書式:

01	[クローラーグループ名], [User-Agent パターン]
02	:

例:

01	crawlerGoogle, *Googlebot-Mobile
02	crawlerYahoo, *Y!J-MBS
03	crawlerYahoo, *Y!J-SRD

- ※ User-Agent パターンの先頭が「*」ではない場合、User-Agent と先頭一致による比較を行います。
- ※ User-Agent パターンの先頭が「*」の場合、User-Agent と「*」より後の部分を部分一致による比較を行います。

4-7 クローラーグループの利用

クローラーグループを利用するには、RALCCrawlerGroupFile ディレクティブでクローラーグループ設定ファイルを指定します。

RALCCrawlerGroupFile ディレクティブは、<Location>ディレクティブ内でも指定可能です。

例:

01	<Location /foo/>
02	RALCCrawlerGroupFile /var/roundabout/conf/crawler-group.csv
03	</Location>

コンテンツ内で、クローラー向けに文言を表示したい場合など、コンテンツの出し分けでクローラーグループを使うこともできます。こちらの利用方法については、第 5 章「[表示コントロール\(PI\)](#)」を参照ください。

4-8 グループ設定の再読み込み

ユーザー端末グループ設定ファイルおよびクローラーグループ設定ファイルは、Apache 起動時に一度だけ読み込まれます。変更した内容を反映させるには Apache を再起動します。

なお、開発などの用途でグループ設定の自動再読み込みを行うには、RALCAutoReload ディレクティブで設定が可能です。自動再読み込み動作中はグループ設定ファイルが更新された場合、変更内容が即座に反映されます。

デフォルト時:

RALCAutoReload Off

自動再読み込み有効時:

RALCAutoReload On

第5章 表示コントロール(PI)

表示コントロール(PI)機能を利用すると、アクセスした端末の端末情報(第3章「[端末情報ファイル](#)」および第4章「[端末グループ](#)」を参照)に従い、コンテンツ内の表示の一部を切り替える事が可能です。例えば、`<?ra g="IPHONE"?>` という文字列を行の先頭に記述し、続けてタグやテキストを記述します。正しい記述内容でない場合、PI が書かれた行すべてが削除されます。

例:

01	<code><p></code>
02	サンプルページです。
03	<code><?ra g="IPHONE"?></code> iPhone からアクセスされました。
04	<code></p></code>

※ 「g」は「Device-Group」の略記です。

アクセスした端末が iPhone であれば、直後に続けたタグやテキストは表示されます。それ以外の端末であれば、直後タグやテキストは表示されません。

HTML 出力例 (iPhone からアクセスした場合):

01	<code><p></code>
02	サンプルページです。
03	iPhone からアクセスされました。
04	<code></p></code>

HTML 出力例 (Android 端末からアクセスした場合):

01	<code><p></code>
02	サンプルページです。
03	<code></p></code>

※ ダブルクォーテーション内の値は、大文字・小文字が区別されます。

5-1 利用可能な端末情報

端末情報ファイル (terminfo.csv) と端末グループ (device-group.conf) の内容が利用できます。端末グ

ループを使うには、g="[端末グループ名]" と書きます。端末グループの詳細については、第 4 章「端末グループ」を参照ください。端末情報については、第 10 章「付録」の「端末情報ファイルの内容」を参照ください。

5-2 PI で使用できる書式

(1) 一致

書式:

01	<?ra [端末情報項目名]=" [値]"?>切替対象の文字列
----	---------------------------------

例:

01	<?ra Carrier="DoCoMo"?>この端末はドコモです
----	-----------------------------------

アクセスした端末のスペック値が指定した項目の「値」と一致する場合、PI 以後の文字列が残されます。「値」と一致しない場合、その一行は削除されます。上記の例では、ドコモ端末からアクセスした場合は「この端末はドコモです」と表示され、それ以外の端末では何も表示されません。

(2) 範囲指定

書式:

01	<?ra [端末情報項目名]=" [開始値]-[終了値]"?>切替対象の文字列
----	---

例:

01	<?ra Browser-Width ="1-240"?>この端末は QVGA 以下の解像度です
----	--

アクセスした端末のスペック値が指定した項目の範囲内に含まれる場合、PI 以後の文字列が残されます。範囲から外れた場合、その一行は削除されます。上記の例では、ブラウザ幅が 240 ピクセル以下の端末からアクセスした場合は「この端末は QVGA 以下の解像度です」と表示され、それ以外の端末では何も表示されません。

(3) 否定

書式:

01	<?ra [端末情報項目名]=" ![値]"?>切替対象の文字列
----	----------------------------------

例:

01 <?ra Carrier="!DoCoMo"?>ドコモ以外の端末です

アクセスした端末のスペック値が指定した項目の「値」と一致しない場合、PI 以後の文字列が残されます。「値」と一致した場合、その一行は削除されます。上記の例では、ドコモ以外の端末からアクセスした場合は「ドコモ以外の端末です」と表示され、ドコモ端末からアクセスした場合は何も表示されません。

否定条件は属性値の先頭にのみ記述でき、属性値全体にかかります。

(4) OR

書式:

01 <?ra [端末情報項目名]=" [値 1] | [値 2] | [値 3] | ... "?>切替対象の文字列

例:

01 <?ra Maker="LG|Motorola|NOKIA|PANTECH|SAMSUNG"?>海外端末です

アクセスした端末のスペック値が指定した項目のいずれかの「値」と一致する場合、PI 以後の文字列が残されます。どの「値」とも一致しなかった場合、その一行は削除されます。上記の例では、NOKIA やサムスンなど海外メーカーの端末からアクセスした場合は「海外端末です」と表示され、それ以外の端末からアクセスした場合は何も表示されません。

※ 否定条件の場合は、属性値全体を否定にします。部分的に否定条件を使うことはできません。

(5) AND

書式:

01 <?ra [端末情報項目名 1]=" [値 1]" [端末情報項目名 2]=" [値 2]" ... ?>切替対象の文字列

例:

01 <?ra Carrier="SoftBank" Screen-Width="480"?>SoftBank の VGA 端末です

アクセスした端末のスペック値が指定した項目の「値」とすべて一致する場合、PI 以後の文字列が残されます。いずれかの「値」に一致しなかった場合、その一行は削除されます。上記の例では、SoftBank で液晶サイズが幅 480 ピクセルの端末からアクセスした場合は「SoftBank の VGA 端末です」と表示され、それ以外の端末からアクセスした場合は何も表示されません。

また、同じ端末情報項目名を複数回使うことはできません。

(6) ユーザー拡張項目へのアクセス方法

ユーザー拡張項目は、端末情報ファイルの「User-Extention(略記:ue)」項目のことです。この項目は「A02110」のような値を持ち、各桁にそれぞれ意味があります。各桁のデータを取得するには「ue.1」のように項目名と「n個目」の数字をドット「.」でつなぎます。「n個目」は左から1、2、3…と続き、1～34の間で指定可能です。

書式:

01	<?ra ue. [n 個目]=" [値]" ?>切替対象の文字列
----	-----------------------------------

例:

01	<?ra ue. 3="2">左から3個目の値が「2」の端末からのアクセスです
----	---

※ 範囲外の値や存在しない値が指定された場合は、PIの書かれた行すべてが削除されます。

(7) 複数行にわたる PI

PIにline属性を追加することで、PIの書かれた行を含む、それ以降の複数行を出し分けの対象とすることができます。

書式:

01	<?ra [端末情報項目名]=" [値]" line=" [行数]" ?>切替対象の文字列
02	切替対象の文字列
03	:

例:

01	<?ra g="SP" line="4">スマートフォン用のコンテンツ
02	<div id="A">
03	JavaScript のコンテンツ
04	</div>

lineの行数を満たす前に次のPIが出現した時点で、前のPIは終了し、新しいPIが有効になります。lineの値を「*(アスタリスク)」とすると、行数制限なしになります。<?ra break?> が出現すると、その時点で前のPIは終了します。<?ra break?> の書かれた行は削除されます。

書式:

01	<?ra [端末情報項目名]="[値]" line="10"?>切替対象の文字列
02	切替対象の文字列
03	<?ra [端末情報項目名]="[値]" line ="*"?>切替対象の文字列
04	切替対象の文字列
05	<?ra break?>

例:

01	<?ra g="SP" line="10"?>スマートフォン用のコンテンツ
02	<div id="A">
03	JavaScript のコンテンツ
04	</div>
05	<?ra g"!SP" line="*"?>携帯用のコンテンツ
06	<div>
07	静的コンテンツ
08	</div>
09	<?ra break?>

※ 複数行 PI はブロック形式ではありません。複数の PI の条件が同時に有効になることはありません。

(8) タグ表記

PI 形式以外に空要素用のタグ形式で表記することもできます。PHP で「<?~?>」の記述形式がバッティングしてしまう場合は、こちらのタグ形式をお使いください。

書式:

01	<ra:pi [端末情報項目名 1]="[値 1]" [端末情報項目名 2]="[値 2]" ... />切替対象の文字列
----	---

例:

01	<?xml version="1.0" encoding="UTF-8"?>
02	<html xmlns="http://www.w3.org/1999/xhtml" xmlns:ra="http://www.symmetric.co.jp/roundabout/">
03	<ra:pi Carrier="SoftBank" Screen-Width="480" />SoftBank の VGA 端末です

- ※ 名前空間 URI は <http://www.symmetric.co.jp/roundabout/> です。(2 行目)
- ※ 名前空間プレフィックスは ra 固定です。(4 行目)
- ※ 空要素タグ形式のみをサポートします。
- ※ 複数行 PI に使う `<?ra break?>` のタグ表記は「`<ra:pi break="break" />`」です。

第6章 テーブル変換

ラウンドアバウト 2.2.0 から、docomo(foma1x, foma2x)、au(フィーチャーフォン)の端末に対して、float などを利用した表現をテーブルレイアウトに変換する機能が使用できます。

docomo(foma1x, foma2x)、au(フィーチャーフォン)の端末は float が効かないため本来は表示が崩れてしまいますが、テーブルに変換することで期待したレイアウトに近い表現を実現できます。

本機能を使用するには、変換したい要素に「ra:cols」属性を付加する必要があります。

書式:

01	<変換対象のタグ ra:cols="列数 変換タイプ">
02	・・・(変換対象の子要素)
03	</変換対象のタグ>

列数: 列数には table 化した時の列を指定します。列数に足りない場合は、自動で最後の列を拡張して (colspan 指定)合わせます。

変換タイプ: 変換タイプにはタイプ1～タイプ3があり、それぞれ変換ルールが異なります。詳細は以下の節を参照してください。

6-1 タイプ1(リスト表現)

タイプ1は図例のようなリストの子要素を float で横並びにした表現をテーブルレイアウトに変換します。具体的には ra:cols を指定した要素の、第 1 子要素を tr タグに変換し、その各子要素を td タグで囲う変換を行います。

図例: タイプ1(リスト表現)

2012/05/16		同社では社員2万人に全員 iPhone・iPadを配布し、シンクラ...
2012/05/14		KDDIは2012年度を「モバイルワークスタイル元年」と位置...

例: コード(変換前)

01	<ul ra:cols="3 type1" class="bar">
----	------------------------------------

```

02 <li class="foo">
03   <p>2012/05/16</p>
04   
05   <div>
06     同社では社員 2 万人に全員 iPhone・iPad を配布し、シンクラ…
07   </div>
08 </li>
09 <li class="foo">
10   <p>2012/05/14</p>
11   <div>
12     KDDI は 2012 年度を「モバイルワークスタイル元年」と位置…
13   </div>
14 </li>
15 </ul>

```

例:コード(変換後)

```

01 <table class="bar">
02   <tr class="foo"><td class="_bar_td1">
03     <div>2012/05/16</div><br />
04     </td><td class="_bar_td2">
05     </td><td class="_bar_td3"><div>
06       同社では社員 2 万人に全員 iPhone・iPad を配布し、シンクラ…
07     </div>
08   </td></tr>
09   <tr class="foo"><td class="_bar_td1">
10     <div>2012/05/14</div><br />
11     </td><td colspan="2" class="_bar_td2"><div>
12       KDDI は 2012 年度を「モバイルワークスタイル元年」と位置…
13     </div>
14   </td></tr>
15 </table>

```

6-2 タイプ2(メニュー表現)

タイプ2は図例のようなリストの各要素を float で横並びにした表現をテーブルレイアウトに変換します。主にメニューなどのレイアウトに使用します。

具体的には `ra:cols` を指定した要素の子要素を `td` タグに変換して変換を行います。`tr` タグは列数で指定した列に達したタイミングで追加されます。

図例: タイプ2 (メニュー表現)

HOME	病院概要	看護部
職員募集	アクセス	電話をかける

例: コード (変換前)

```

01 <ul class="menu" ra:cols="3 type2">
02   <li class="active"><a href="xxxx.html">HOME</a></li>
03   <li><a href="xxxx.html">病院概要</a></li>
04   <li><a href="xxxx.html">看護部</a></li>
05   <li><a href="xxxx.html">職員募集</a></li>
06   <li><a href="xxxx.html">アクセス</a></li>
07   <li><a href="xxxx.html">電話をかける</a></li>
08 </ul>

```

例: コード (変換後)

```

01 <table class="menu">
02   <tr><td class="active"><a href="xxxx.html">HOME</a></td>
03   <td><a href="xxxx.html">病院概要</a></td>
04   <td><a href="xxxx.html">看護部</a></td></tr>
05   <tr><td><a href="xxxx.html">職員募集</a></td>
06   <td><a href="xxxx.html">アクセス</a></td>
07   <td><a href="xxxx.html">電話をかける</a></td></tr>
08 </table>

```

6-3 タイプ3 (回り込み表現)

タイプ3は図例のような、画像へのテキストの回り込みを `float` で表現したものをテーブルレイアウトに変換します。

具体的には `ra:cols` を指定した要素を `table` タグに変換し、その子要素の前に `td` タグを挿入します。

図例: タイプ3 (回り込み表現)



プーケット - だれもが憧れる夢の島

絶対納得の価格。会社をズル休みしてでも行きたい憧れのビーチ。なにもしないが贅沢...

7泊9日プラン
109,900円～

例: コード (変換前)

```

01 <article class="main_plan clear" ra:cols="2 type3">
02   <div class="thumbnail fleft">
03     
04   </div>
05   <div>
06     <a href="prod.html">
07       <h3 class="main_plan_title">プーケット - だれもが憧れる夢の島</h3>
08       <p class="main_plan_disc">絶対納得の価格。会社をズル休みしてでも行きたい憧れのビ
09       ーチ。なにもしないが贅沢...</p>
10       <p class="taright">
11         <span class="main_plan_price_sub"> 7 泊 9 日プラン</span>
12         <span class="main_plan_price"> 109,900 円～ </span>
13       </p>
14     </a>
15   </div>
16 </article>

```

例: コード (変換後)

```

01 <table class="main_plan clear">
02   <tr><td class="_main_plan_td1"><div class="thumbnail fleft">
03     
04   </div></td>
05   <td class="_main_plan_td2"><div>
06     <a href="prod.html">
07       <span class="main_plan_title">プーケット - だれもが憧れる夢の島</span><br />

```

08	<code>絶対納得の価格。会社をズル休みしてでも行きたい憧れの</code>
	<code>ビーチ。なにもしないが贅沢...
</code>
09	<code></code>
10	<code> 7泊9日プラン</code>
11	<code> 109,900 円~ </code>
12	<code>
</code>
13	<code></code>
14	<code></div></td></tr></code>
15	<code></table></code>

6-4 補足

(1) a タグ内のブロック要素変換

ra:cols 属性を指定した要素に含まれる a タグ内の block 要素は、フィーチャーフォンでは span タグに変換されます。

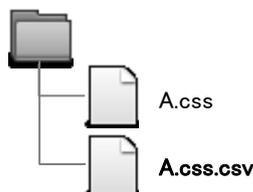
- 変換対象となるタグは address、blockquote、div、h1~h6、p です。
- 終了タグの後には
を挿入します。
ただし、以後終了タグしか存在しない場合には挿入しません。

第7章 CSS 変換シート

機種によって最適値が異なるプロパティにおいて、その値を端末グループごとに出し分けることができるのが CSS 変換シートです。例えば、解像度の違う端末では、同じ値のマージンでも、違った空き具合に見えます。このような、若干の違いをコントロールしたい場合に、CSS のプロパティ値をピンポイントで変換できるのが CSS 変換シートです。CSS 変換シートのファイル形式は CSV です。

7-1 CSS 変換シートの配置場所

- ファイル名は「対象 CSS ファイル名.css.csv」とします。
- 変換対象となる CSS ファイルと同じディレクトリに配置します。



7-2 CSS 変換シートの内容

ファイルの内容は以下のようになります。実際のファイルは、「template/css/style.css.csv」を参考にしてください。

selector_name	property_name	property_value	端末グループ A	端末グループ B	端末グループ C
	font-size	medium	x-small		large
	font-size	small	xx-small		medium
p	line-height	1.2em		1.5em	1.0em
div .class1	margin		5px	10px	8px

- selector_name・・・変換したい値のセレクタを書きます。何も指定しない場合は、変換対象となる CSS ファイル内のすべてのセレクタが対象となります。
- property_name・・・変換したい値のプロパティ名を書きます。selector_name が省略された同名のプロパティ・値がある場合、上に書かれたものが適用されます。この項目は省略できません。
- property_value・・・変換したい値を書きます。何も指定しない場合は、CSS ファイルに書かれた内容がそのまま出力されます。

- 端末グループを増やすには、列を追加してください。複数の端末グループに属する端末は、左に書かれたグループの内容が適用されます。

※ ここで使うグループの内容は排他ではありません。1つの端末が複数のグループに属することもあります。

変換対象 CSS ファイル例:

```
01 body { font-size: medium; }
02 p { line-height: 1.2em; font-size: small; }
03 div.class1 { margin: 10px; }
```

端末グループ A と B に属する端末が読み込む CSS:

```
01 body { font-size: x-small; }
02 p { line-height: 1.2em;
03     font-size: xx-small; }
04 div.class1 { margin: 5px; }
```

※ 端末グループ A の値が適用されます。

端末グループ B と C に属する端末が読み込む CSS:

```
01 body { font-size: medium; }
02 p { line-height: 1.5em;
03     font-size: small; }
04 div.class1 { margin: 10px; }
```

※ 端末グループ B の値が適用されます。

CSS 変換が働くと、変換されたプロパティ値を含むセレクタ全体を、CSS 内部参照として HTML 内に展開します。

複数の端末グループに属している場合は、CSS 変換シートの左側の値が適用されます。

端末グループの設定については、第 4 章「[端末グループ](#)」を参照ください。

7-3 CSS 変換ルール適用の優先度

1. セレクタ + プロパティ名 + プロパティ値
2. プロパティ名 + プロパティ値
3. セレクタ + プロパティ名
4. プロパティ名

第8章 CSS・JavaScript ファイル切り替え

コンテンツによっては、とある端末グループには全く別の CSS ファイルを専用に用意するほうが効率がよかったり、一部読み込ませたくないプロパティなどがあって同じ CSS ファイルだとうまくいかなかったりと、CSS ファイル自体を切り替えたいというときがあります。同様に、JavaScript ファイルも端末グループごとに JavaScript ファイルを切り替えて使用したいときがあります。

ra:rename 属性を使うと、端末グループごとに link タグを書き換えて、読み込む CSS ファイルを切り替えることができます。また、特定の端末グループ以外は CSS ファイルを参照しないように指定することもできます。

ra:rename 属性は、link タグや script タグの属性として指定します。

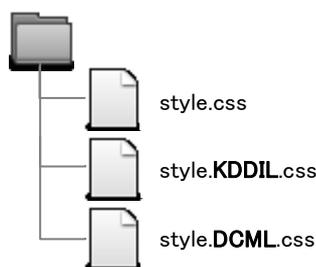
8-1 CSS ファイル切り替えの例

au ブラウザ 6 系と、i モードブラウザ 1.0 端末には、style.css とは異なる CSS を読ませたい場合には、次のように ra:rename 属性を指定します。

ra:rename 属性の内容: ※rel、type 属性は省略しています。

```
01 <link href="style.css" ra:rename="*:KDDIL, DCML" />
```

用意するファイル:



- style.css … CSS を出し分ける必要のない端末に読ませたい CSS ファイル。
- style.KDDIL.css … au ブラウザ 6 系端末用の CSS ファイル。
- style.DCML.css … i モードブラウザ 1.0 端末用の CSS ファイル。

変換結果:

出し分けが必要のない端末は、link タグが以下のように書き換えられます。

```
01 <link href="style.css" />
```

KDDIL グループに属する端末は、link タグが以下のように書き換えられます。

```
01 <link href="style.KDDIL.css" />
```

DCML グループに属する端末は、link タグが以下のように書き換えられます。

```
01 <link href="style.DCML.css" />
```

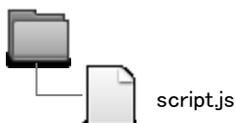
8-2 JavaScript ファイル切り替えの例

スマートフォンのみ外部 JavaScript ファイルを使用したい場合には、次のように ra:rename 属性を指定します。

ra:rename 属性の内容:

```
01 <script type="text/javascript" src="script.js" ra:rename="SP"></script>
```

用意するファイル:



・script.js … スマートフォン用の JavaScript ファイル。

変換結果:

SP グループ(スマートフォン)に属する端末は、script タグが以下のように書き換えられます。

```
01 <script type="text/javascript" src="script.js"></script>
```

それ以外の端末では script タグが HTML から削除されます。

※ フィーチャーフォンのみ script タグを削除する別の方法として、RALCRemoveScript ディレクティブを使用する方法もあります。詳しくは設定の詳細は、別紙「Apache 設定リファレンス」の RALCRemoveScript 項を参照ください。

8-3 ra:rename 属性の書式

コロン「:」とカンマ「,」を使って、端末グループ名を指定します。コロンの左側に書かれたグループには、href 属性もしくは src 属性の内容をそのまま出します。右側に書かれたグループには、href 属性もしくは src 属性の値を [ファイル名].[端末グループ名].[拡張子] 形式に書き換えます。コロンの右側は、カンマ「,」で区切って複数の端末グループを指定できます。

指定する端末グループは必ずしも排他ではないので、コロン「:」の右側を左から評価していき、最初にヒットしたグループで href 属性もしくは src 属性の内容を書き換えます。コロンの左側は最後に評価されます。ra:rename 属性に指定されたグループのいずれにも当てはまらなければ、link 要素もしくは script 要素は削除されます。

書式:

```
01 <link href="style.css" ra:rename="[端末グループA]:[端末グループB],[端末グループC]" />
```

結果:

端末グループ B に属する端末は、link タグが以下のように書き換えられます。

```
01 <link href="style.B.css" />
```

端末グループ C に属する端末は、link タグが以下のように書き換えられます。

```
01 <link href="style.C.css" />
```

端末グループ A に属する端末は、link タグが以下のように書き換えられます。

```
01 <link href="style.css" />
```

端末グループに属さない端末は、link タグ削除が削除されます。

コロンの左側には、アスタリスク「*」を指定することができます。これは、コロンの右側のグループ以外の全てのグループに href の内容をそのまま出したい場合に使います。

書式:

```
01 <link href="style.css" ra:rename="*:[端末グループB],[端末グループC]" />
```

結果:

端末グループ B に属する端末は、link タグが以下のように書き換えられます。

```
01 <link href="style.B.css" />
```

端末グループ C に属する端末は、link タグが以下のように書き換えられます。

```
01 <link href="style.C.css" />
```

上記以外の端末グループに属する端末は、link タグが以下のように書き換えられます。

```
01 <link href="style.css" />
```

ra:rename 属性に、1 つの端末グループだけを指定した場合は、指定のグループのときのみ link 要素を出します。

書式:

```
01 <link href="style.css" ra:rename="[端末グループA]" />
```

結果:

端末グループ A に属する端末は、link タグが以下のように書き換えられます。

```
01 <link href="style.css" />
```

上記以外の端末グループに属する端末は、link タグ削除が削除されます。

第9章 HTML 変換シート

ラウンドアバウトは、HTML 変換シート (conversion.csv) に記載されている変換ルールによって、コンテンツ変換を行います。出荷時に提供される変換ルールには、公式サイトで公開されている仕様に従ったルールが含まれ、3 キャリア対応コンテンツの制作が可能な状態にあります。

また、デフォルトのルールを編集、もしくは、独自のルールを追記することで、より自由度の高い変換ルールを構築することが可能です。

9-1 HTML 変換シートの種類

(1) デフォルト言語変換シート

XHTML もしくは HTML のコンテンツに対して言語変換を行う変換シートです。XHTML、および HTML の判断は Content-Type により行われます。この変換シートのフォーマットは CSV 形式です。

(2) 絵文字変換シート

XHTML もしくは HTML のコンテンツに対して絵文字変換を行う変換シートです。XHTML、および HTML の判断は、Content-Type により行われます。この変換シートの書式は、デフォルト言語変換シートと同一です。

(3) 独自言語変換シート

XHTML もしくは HTML のコンテンツに対して独自の言語変換を行う変換シートです。XHTML、および HTML の判断は Content-Type により行われます。この変換シートのフォーマットは CSV 形式です。

9-2 HTML 変換シートの書式

HTML 変換シートは CSV 形式で提供されます。一部、エクセルで提供される場合があります。

エンコーディングを UTF-8 または Shift_JIS としてください(※)。

1 行目はコメントで書式のバージョンを示します。

2 行目はヘッダ行です。

1 列目はキーワードです。

2 列目以降は端末グループ名です。

3 行目以降はキーワード(変換対象文字列)と変換後文字列を列挙します。各列には RFC8140 準拠の文字列が指定できます。

3 行目以降に変換後文字列として「;」を指定した場合、変換後の文字列が変換対象文字列(1 列目)と同一とみなされます。なお、「;」に変換を行いたい場合は、「;;」と指定します。

変換は対象と完全一致の場合のみ処理されます

※ RALCSourceEncoding が UTF-8 かつ both オプション付で設定されている場合(デフォルト設定)、変換シートの文字エンコーディングは UTF-8 とし、それ以外の設定の場合(both オプションなしの UTF-8 設定や Shift_JIS, EUC-JP で設定している)は Shift_JIS の変換シートを使用してください。詳細は別紙「Apache 設定リファレンス」を参照ください。

9-3 HTML 変換シートにおける端末グループ

HTML 変換シートで利用する端末グループは、各キャリア言語仕様の違いや端末スペックによって分類された、端末情報ファイルで定義されているグループです。1 つの端末に 1 つだけ与えられた完全に排他のグループになります。

システム端末グループ名	意味
foma3x	i モードブラウザ 2.x 搭載端末
foma2x	i モードブラウザ 1.0 を搭載し、テーブル関連タグ対応端末
foma1x	i モードブラウザ 1.0 を搭載し、テーブル関連タグ非対応の端末
pd2	mov a で、画面解像度が QVGA の端末
pd1	mov a で、画面解像度が QVGA 未満のもの
win	WIN 端末
cdma	au の Axxxx、Cxxxx シリーズ
sb3g	SoftBank3G
will	WILLCOM
em	emobile
iphone	iOS 搭載端末、Android 端末
pc	PC

※ システム端末グループを PI などを利用する場合には、端末グループ設定ファイル (device-group.conf) で、このグループを再定義していますので、そちらを使ってください。詳細は第 10 章「付録」の「初期設定済のユーザー端末グループ」を参照ください。

9-4 独自言語変換シートの作成・編集

独自言語変換シートは /var/roundabout/conf/ 以下に original-conversion.csv などの名前で CSV フ

ファイルを作成し、それを編集することで独自言語変換シートになります。列項目はデフォルト言語変換シートと同じ内容にしてください。列項目については、同章 8-3「HTML 変換シートにおける端末グループについて」の表内「システム端末グループ」を参照ください。作成・編集には Microsoft Excel を利用すると非常に便利です。但し、一部のルール(全角数字など)では、Microsoft Excel のオートコレクト機能で修正されてしまうものがありますので若干の注意が必要です。

※ 文字エンコーディングは、デフォルト言語変換シート同様、RALCSourceEncoding の設定に応じて適切なエンコーディングで作成してください。

9-5 HTML 変換シートの適用

HTML 変換シート(デフォルト・独自)は、roundabout-vhost.conf の RALCConversionSheetFile、RALCGlyphConversionSheet ディレクティブで指定します。変換シートは、ディレクトリ別に設定することができますその設定は、同ファイルの Location ディレクティブで指定します。独自言語変換シートを適用する場合はデフォルト変換シートの適用記述に加え、独自言語変換シートの適用記述を行ってください。Location ディレクティブ内では適用したい変換シートをすべて記述してください。適用した複数変換シート内で変換ルールが同じものは、より後述の変換ルールが適用されます。

デフォルト言語変換シートのみ例:

01	<Location /foo/>
02	RALCConversionSheetFile /var/roundabout/conf/conversion-utf8.csv
03	RALCGlyphConversionSheetFile /var/roundabout/conf/glyph-conversion-utf8.csv
04	RALCSourceEncoding UTF-8 both
05	</Location>

デフォルト言語変換シート+独自言語変換シートの例:

01	<Location /foo/>
02	RALCConversionSheetFile /var/roundabout/conf/conversion-utf8.csv
03	RALCConversionSheetFile /var/roundabout/conf/original-conversion-utf8.csv
04	RALCGlyphConversionSheetFile /var/roundabout/conf/glyph-conversion-utf8.csv
05	RALCSourceEncoding UTF-8 both
06	</Location>

9-6 HTML 変換シートの再読み込み

HTML 変換シートはデフォルト設定では、Apache 起動時に一度だけ読み込まれます。シートの内容を変更した場合は Apache の再起動後に反映されます。

なお、開発などの用途で HTML 変換シートの自動再読み込みを行うには、RALCAutoReload ディレクティブで設定が可能です。自動再読み込み動作中は HTML 変換シートが更新された場合、変更内容が即座に反映されます。

デフォルト時:

RALCAutoReload Off

自動再読み込み有効時:

RALCAutoRelead On

第10章 キャラクタエンコーディング変換

ラウンドアバウトには XHTML/HTML コンテンツのキャラクタエンコーディング変換機能があります。キャラクタエンコーディング変換機能は、レスポンスボディのキャラクタエンコーディング変換処理とレスポンスヘッダ Content-Type ヘッダの変換処理とで構成されます。

10-1 変換対象のキャラクタエンコーディング

ラウンドアバウトが対応しているキャラクタエンコーディングは、以下の通りです。

- UTF-8
- Shift_JIS
- EUC-JP

10-2 使用可能な文字セット

使用可能な文字セットは、Windows-31J と絵文字です。具体的には以下の文字セットです。

- JIS X 0201:1997 (ラテン文字・カタカナ)
- JIS X 0208:1997 (第 1・第2水準漢字)
- NEC 特殊文字 (丸つき数字など)
- NEC 選定 IBM 拡張文字 (「高」など)
- IBM 拡張文字 (「高」など)
- i モード対応絵文字 (基本絵文字及び拡張絵文字)
- SoftBank 絵文字

以上の文字セットに含まれない文字には対応していません。

10-3 Content-Type 変換

ラウンドアバウトでは、RALCSourceEncoding の設定と端末グループにより、全ての HTML/XHTML コンテンツを UTF-8 または Shift_JIS で出力します。そのため、Content-Type ヘッダの charset パラメータを出力文字エンコーディングと同じに設定します。

RALCSourceEncoding	システム端末グループ名	出力文字エンコーディング
UTF-8 (both オプションあり)	cdc1, cdc2, win, cdma, will	Shift_JIS
	上記以外	UTF-8
UTF-8, Shift_JIS, EUC-JP (both オプションなし)	全グループ	Shift_JIS

10-4 キャラクタエンコーディングの指定

ラウンドアバウトではキャラクタエンコーディングを RALCSourceEncoding ディレクティブにて明示的に設定します。RALCSourceEncoding ディレクティブが未設定の場合は、Shift_JIS が使用されます。なお、Content-Type ヘッダの charset パラメータや XML 宣言のエンコーディング指定、meta タグの Content-Type 指定値は無視され、内容による自動判別も行われません。

例:

01	<Location /foo/>
02	RALCSourceEncoding UTF-8 both
03	</Location>

RALCSourceEncoding ディレクティブには、UTF-8・Shift_JIS・EUC-JP のいずれかが指定可能です。また、UTF-8 指定のみ both オプションを指定可能です。both オプションを指定した場合、フォームからの GET/POST 送信時、UTF-8 アプリケーション側へ送信されるリクエストの文字エンコーディング変換に対応します。

第11章 付録

11-1 端末情報ファイルの内容(terminfo2.csv)

※ 数値の項目は範囲指定(第5章「表示コントロール(PI)」参照)が使えます。大文字・小文字が区別されますので注意してください。略称があるものは、そちらもPIなどで利用できます。

※ 2011年5月24日現在の値です。

項目名 (略称)	意味	値	値の意味
Carrier (cr)	キャリア	<文字列> DoCoMo、SoftBank、Disney、 au、WILLCOM、emobile、 Apple、PC	
Device-Name	端末名	<文字列> 例:SH-09C、T006 (※1)	
Cache-Size (cs)	ブラウザキャッシュサイズ	<数値> 単位:byte 例:512000 (※1)	
Browser-Width (bw)	ブラウザ横ピクセル数	<数値> 単位:px 例:471 (※1)	
Browser-Height (bh)	ブラウザ縦ピクセル数	<数値> 単位:px 例:715 (※1)	
Screen-Width (sw)	待受画面横ピクセル数	<数値> 単位:px 例:480 (※1)	
Screen-Height (sh)	待受画面縦ピクセル数	<数値> 単位:px 例:854 (※1)	
Flashlite-Version (fv)	Flash Lite バージョン	<数値> 1.0、1.1、2.0、3.0、3.1、4.0 0:非対応	
Screen-Inch	液晶のインチサイズ	<数値> 単位:inch 例:4.0 (※1)	
Browser-Type (bt)	ブラウザ種別	<数値>	
		100	iモードブラウザ1系
		101	iモードブラウザ2系
		102	iモードブラウザ2LE系
		120	auブラウザ6.0系

Browser-Type (bt)	ブラウザ種別	121	au ブラウザ 6.2 系
		122	au ブラウザ 7.2 系
		140	vodafone ブラウザ系
		141	softbank ブラウザ 1.0-Teleca-Browser 系
		142	softbank ブラウザ 1.0-Net-Front 系
		143	softbank ブラウザ 2.0-Net-Front 系
		160	emobile ブラウザ Obigo 系
		161	emobile ブラウザ NetFront 系
		162	emobile ブラウザ OpenBrowser 系
		180	WILLCOM ブラウザ CNF 系
		181	WILLCOM ブラウザ Opera 系
		182	WILLCOM ブラウザ NetFront 系
		200	iPhone Safari
		201	Android Safari
		300	BlackBerry
		400	Opera Mini
		401	Opera Mobile
		500	Internet Explorer Mobile
		600	Firefox for Mobile
		900~999	その他のブラウザ
Browser-Version (bv)	ブラウザバージョン	<数値>	
		1.01	i モード HTML1.0
		1.02	i モード HTML2.0
		1.03	i モード HTML3.0
		1.04	i モード HTML4.0
		1.05	i モード HTML5.0
		1.10	i モード HTML4.0&XHTML1.0
		1.11	i モード HTML5.0&XHTML1.1
		1.20	i モード HTML6.0&XHTML2.0
		1.21	i モード HTML7.0&XHTML2.1
		1.22	i モード HTML7.1&XHTML2.2
		1.23	i モード HTML7.2&XHTML2.3
		2.0	i モードブラウザ 2.0
		2.1	i モードブラウザ 2.1

Browser-Version (bv)	ブラウザバージョン	7 / 7.1 / 7.2 / 8.2 / 8.3	au ブラウザバージョン 6.0 系の詳細なバージョン		
		5.0 / 6.2 / 7.3 / 8.0 / 9.0 / 9.1 / 9.2 / 10.2 / 10.3 / 10.4 / 11.1 / 11.2 / 12.1 / 13.1 / 13.2 / 13.4 / 14.1 / 14.2 / 14.3 / 15.1 / 15.2 / 16.1 / 17.1 / 17.2	au ブラウザバージョン 6.2 系の詳細なバージョン		
		1.0 / 1.3 / 1.5 / 2 / 3 / 4 / 5 / 6	au ブラウザバージョン 7.2 系の詳細なバージョン		
		0.0	Vodafone 系		
		3.3 / 3.4 / 3.5	SoftBank1.0 系 Net-Front		
		3.1	SoftBank1.0 系 Teleca		
		3.5	SoftBank2.0 系		
		2.0	CNF 系		
		7.0 / 7.2	Opera		
		3.3 / 3.4	NetFront 系		
		419.3 / 525.20 / 531.21	iPhone Safari		
		525.20 / 530.17 / 533.1 / 534.13	Android Safari		
		Os-Type (ot)	OS 種別	<数値>	
				0	フィーチャーフォン
100	iOS				
200	Android				
300	BlackBerry				
400	Windows				
500	Symbian				
999	PC の OS				
Os-Version (ov)	OS バージョン	<数値>			
		0: フィーチャーフォン、PC 例: 2.2 (※1)			
Download-Speed	下り速度	<数値> 単位: Kbps 例: 7200 (※1)			

User-Extension (ue)		<ビットアクセス> 例:B04110	
	動画サポート (左から1番目)	0~F 例:3gp + Wmv + Asf = B	
		8	3gp に対応
		4	3g2 に対応
		2	wmv に対応
		1	asf に対応
		0	全て非対応
	スマートフォン動画サ ポート (左から2番目)	0~3 例:mp4 + flv = 3	
		2	mp4 に対応
		1	flv に対応
		0	全て非対応、フィーチャーフォン
	入力インターフェース (左から3番目)	0~F 例:QWERTY キー + タッチ パネル = C	
		8	QWERTY キー搭載
		4	タッチパネル搭載
		2	テンキー搭載
		1	タッチペン付属
table タグ対応 (左から4番目)	1:対応 0:非対応		
GPS 機能 (左から5番目)	1:対応 0:非対応		
JavaScript 対応 (左から6番目)	1:対応 0:非対応 (※2)		
Release-Date	発売年月	<数値> 例:201102 (2011年02月)	

※1 … 詳細は端末情報ファイル参照。

※2 … ※対応(1)となる機種は以下の通り。

- ・iモードブラウザ 2.0 以上
- ・SoftBank2.0 機種以上(500KB ブラウザ)
- ・スマートフォン
- ・PC

11-2 初期設定済のユーザー端末グループ (device-group.conf)

グループ名	説明
SP	スマートフォン全般
IPHONE	iPhone
ANDROID	Android 端末
WPHONE	WindowsPhone 端末 ※バージョン 2.1.0 以降の初期ファイルにて定義
TABLET	7 インチ(GALAXY TAB)以上の画面サイズを持つ端末
DCM	i モードブラウザ 2.x 搭載の docomo 端末
DCML	i モードブラウザ 1.0 搭載の docomo 端末
SBM	画面解像度が VGA の SoftBank 端末
SBML	画面解像度が QVGA の SoftBank 端末
KDDI	ブラウザバージョン 7.2 系の au 端末
KDDIL	ブラウザバージョン 6.x 系の au 端末
OTHER3G	3 キャリア以外のキャリア
MB	携帯(フィーチャーフォン)全般 ※バージョン 2.1.0 以降の初期ファイルにて定義
DGFOMA3	i モードブラウザ 2.x 端末 (端末情報ファイルのグループ再定義)
DGFOMA2	i モードブラウザ 1.0 を搭載し、テーブル関連タグ対応端末 (端末情報ファイルのグループ再定義)
DGFOMA1	i モードブラウザ 1.0 を搭載し、テーブル関連タグ非対応の端末 (端末情報ファイルのグループ再定義)
DGPDC2	mova で、画面解像度が QVGA の端末 (端末情報ファイルのグループ再定義)
DGPDC1	mova で、画面解像度が QVGA 未満のもの (端末情報ファイルのグループ再定義)

DGWIN	WIN 端末（端末情報ファイルのグループ再定義）
DGCDMA	au の Axxxx、Cxxxx シリーズ（端末情報ファイルのグループ再定義）
DGSB3G	SoftBank3G（端末情報ファイルのグループ再定義）
DGWILL	WILLCOM（端末情報ファイルのグループ再定義）
DGEM	emobile（端末情報ファイルのグループ再定義）
DGIPHONE	スマートフォン（端末情報ファイルのグループ再定義）
DGPC	PC（端末情報ファイルのグループ再定義）
L2S	標準より小さいフォントサイズを使うことが推奨される端末
L2M	標準より大きいフォントサイズを使うことが推奨される端末
NOFLT	ブロック要素のフロートができない端末
USDCM	docomo 非対応機種 (docomo902 シリーズ以前のテーブル非対応端末)
USSBM	SoftBank 非対応端末 (Vodafone)
USKDDI	au 非対応端末 (au のキャッシュサイズ 100KB 未満の端末)
USSHORTBRW	ブラウザ小サイズ端末 (ブラウザ横幅 200 ピクセル未満の端末) ※バージョン 2.0.1 以降の初期ファイルにて定義